fm

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,734	05/04/2001	G. Glenn Henry	CNTR: 2023	8086

23669 7590 03/04/2004

JAMES W HUFFMAN
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
GOLE, AMOL V	
ART UNIT	**PAPER NUMBER**
2183	4

DATE MAILED: 03/04/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/849,734	**Applicant(s)** HENRY ET AL.	
	Examiner Amol V. Gole	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 5/4/01, 6/5/02, 11/29/02.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-45 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-45 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 5/4/01 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 2.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-45 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file:

#2: IDS (6/5/02)

#3: Change of Address (11/29/02)

Oath/Declaration

3. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:

The Declaration refers to the application entitled "SEPARATE BTAC BHT USING GSHARE TO GET MULTIPLE PREDICTIONS FOR SAME BTAC BRANCH WITH BRANCH HISTORY" while the application submitted is entitled "SPECULATIVE HYBRID BRANCH DIRECTION PREDICTOR". Please resubmit the Declaration with the proper title.

Specification

Content of Specification

4. Claim or Claims: While there is no set statutory form for claims, the present Office practice is to insist that each claim must be the object of a sentence starting with "I (or we) claim," "The invention claimed is" (or the equivalent). See 37 CFR 1.75 and MPEP § 608.01(m).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> *A person shall be entitled to a patent unless –*
>
> *(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.*

5. Claims **22, 37-39** are rejected under 35 U.S.C. 102(b) as being anticipated by Emma et al. (US005353421A).

6. **In regard to claim 22:**

7. Emma et al. teaches a speculative branch target address cache (BTAC) (fig. 10, BHT 12) in a microprocessor, the BTAC comprising:

an array, configured to store branch instruction direction predictions (fig. 9 shows the organization of the BHT [col. 5, lines 3-5] with an array structure with branch direction predictions stored in the 'T' bits);

an input, coupled to said array, configured to receive an instruction cache fetch address, said fetch address indexing into said array to select one of said direction predictions (fig. 10 shows the fetch address is inputted into the BHT 12 and fig. 11 shows the indexing function in block 102); and

an output, coupled to said array, for providing said one of said direction predictions to a branch control logic (fig. 11 shows that an output from the array has one of the direction predictions ('T' bit) and this is outputted to branch control logic (elements 37, 34, 35) as shown in fig. 12);

wherein the branch control logic causes the microprocessor to speculatively branch if said one of said direction predictions specifies a taken direction (col. 12, lines 6-8 and col. 2, lines 57-62 indicate that the processor will speculatively branch on a prediction of taken because it is at instruction-fetch time when it is not known whether the branch is present in the cache line or not), regardless of whether a branch instruction is present in a line of the instruction cache indexed by said fetch address.

8. **In regard to claim 37:**

9. Emma et al. disclose a method for speculatively branching in a microprocessor (fig. 10), the method comprising:

generating a plurality of speculative branch direction predictions of an instruction (fig. 11 shows that in response to a single instruction fetch, a plurality of speculative branch direction predictions indicated by the 'T' entries in the BHT's segment –entry information are generated);

selecting one of said plurality of speculative branch direction predictions as a final direction prediction (fig. 11 shows that a single direction is selected among the plurality by the select logic and select gates); and

speculatively branching the microprocessor if said final direction prediction indicates said instruction will be taken (col. 12, lines 6-8 indicate that if the prediction is to take to branch, fetch the target address of the branch. As this prediction is during instruction-fetch time, the branching is speculative);

wherein said generating, said selecting, and said speculatively branching are preformed prior to decoding said instruction (col. 5, lines 30-32 and col. 12, lines 6-8).

10. In regard to claim 38:

11. Emma et al. discloses the method of claim 37, further comprising: detecting (that) said final direction erroneously indicated said instruction will be taken subsequent to said speculatively branching (col. 8, 61-63).

12. In regard to claim 39:

Emma et al. disclose the method of claim 38, further comprising: branching to a correct target address in response to said detecting (col. 8, lines 65-68, col. 9, lines 1-2 indicate that the processor is restarted when a misprediction is detected. Although not explicitly mentioned, branching to the correct target address calculated at the execution stage is required in order to restart the processing).

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

14. Claims **1-11** are rejected under 35 U.S.C. 103(a) as being unpatentable over Emma et al. (US005353421A) in view of McFarling ("WRL Technical Note TN-36, Combining Branch Predictors," Digital Equipment Corp., 1993).

15. In regard to claim 1:

16. Emma et al. discloses a branch apparatus within a microprocessor (fig. 10) that utilizes a fetch address to select an instruction in an instruction cache (fig. 10 shows fetch address on line 54 is used to select a instruction in cache 13), the apparatus also using the fetch address to speculatively predict whether a branch instruction will be taken or not taken (the Branch History Table (BHT) 12 is used to predict the outcome of a branch during instruction fetching [col. 4, lines 24-26] by using the fetch address [col.

15, lines 4-5]; the prediction is speculative because it is done in during the instruction fetch stage before it is known whether the instruction is a branch or not [col. 2, lines 57-62]), the branch instruction potentially being present in the instruction cache line (col. 2, lines 57-62), the apparatus comprising:

a first predictor (fig. 10, BHT 12), coupled to the fetch address, for predicting whether the branch instruction will be taken or not taken based on the fetch address (fig. 11);

a logic (fig. 6, Gate 'G', coupled to the fetch address, for providing a binary function ('G' either allows or does not allow the address be fed to the History Array 71) of the fetch address on an output of said logic (fig. 6 shows the details of the Decode History Table (DHT) of fig. 4, col. 14, lines 16-19);

a second predictor (fig. 10, DHT 55), coupled to said logic output (fig. 6), for predicting whether the branch instruction will be taken or not taken based on said output; and

a selector, coupled to the fetch address, for selecting one of said first and second predictors based on the fetch address (As shown in fig. 13, based on the fetch address of the branch instruction in question (BA from decoder), a selection is made on whether the prediction of the BHT is to be used or the DHT by comparing the branch address from the BHT and decoder. If they are not equal, then the DHT is used otherwise the BHT. Hence, although a selector is not explicitly shown, it is deemed inherent in order to perform this function of selection).

17. However, Emma et al. do not explicitly mention that the fetch address is used for selecting a cache **line** in the instruction cache and do not disclose a logic, coupled to the fetch address, for providing a binary function of the fetch address and a global branch history on an output of said logic.

18. "Official Notice" is taken that it is well known and expected in the art to select a cache line from a cache based on a fetch address in order to receive not only the data at the particular address required but the data near to it also to take advantage of the principle of locality while fetching.

19. McFarling teaches that a more efficient prediction might be made using both the branch address and the global history (pg. 9, lines 26-28). He introduces the gshare predictor which uses the exclusive ORing of the branch address and global history to index the history array (pg. 11, lines 31-32) and shows that it has the better prediction capabilities than other global history schemes in most cases (pg. 11, lines 33-35, fig. 11). Also he suggests combining different prediction schemes in order to attain better prediction accuracy (pg. 16, lines 34-35).

20. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor to fetch a cache line in response to a fetch address and further modify the logic presented by Emma et al. by adding exclusive-OR circuitry and a global history storage means wherein the output of the logic is the exclusive-OR (binary function) of the branch (fetch) address and the global branch history.

21. One would have been motivated to do so because by fetching an entire cache line, one can take advantage of the prefetching instructions into the processor before actually addressing them and by using the gshare predictor, it would have improved the prediction accuracy of the prediction means and hence improved the performance of the microprocessor by having less number of branch misprediction stalls.

22. In regard to claim 2:

23. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, wherein said binary function comprises an exclusive OR of at least a portion of the fetch address and said global branch history (pg. 11, lines 31-32).

24. In regard to claim 3:

25. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, wherein said first predictor (BHT 12) is provided by a branch target address cache indexed by the fetch address (Emma: fig. 9 shows the array information of the BHT in which target addresses (TA) of branches are stored).

26. In regard to claim 4:

27. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, wherein said second predictor (McFarling: fig. 10 gshare predictor) is provided by a branch history table indexed by said binary function of the fetch address and said global branch history.

28. In regard to claim 5:

29. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, wherein said selector is provided by a branch target address cache (BHT 12) indexed by the fetch address (Emma: fig. 13 shows that the selection function is provided by the branch address from the BA/TA stack, which is part of the BHT).

30. In regard to claim 6:

31. The combination of Emma et al. in view of McFarling as applied to claim 1 does not explicitly teach that the said selector comprises a bit for selecting between said first and second predictions.

32. However "Official Notice" is taken that it is well known and expected in the art that compare signal information be indicated by a bit such as a compare bit in a status register of a common processor for simplified logic.

33. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to have modified the selector to have a bit indicating the result of the

comparison of the branch addresses in the first and second predictors which is used to select between the first and second predictions.

34. In regard to claim 7:

35. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, wherein each of said first and second predictors comprises a plurality of predictors of whether the branch instruction will be taken or not taken (the first predictor i.e. BHT 12 is shown to comprise of a plurality of predictors as shown in fig. 11 [Emma] indicated by the plurality of 'T' bits; the second predictor consists of a history array/table comprising of a plurality of predictor entries as taught in McFarling pg. 2, lines 35-36), wherein said selector comprises a plurality of bits corresponding to said plurality of predictors, for selecting between corresponding ones of said plurality of first and second predictors (Although not explicitly shown it is deemed inherent that the select logic 105 of the first predictor shown in fig. 11 [Emma] comprises of a plurality of bits to select among plurality of predictions because a single bit cannot address a plurality of predictions. Fig. 10 of McFarling shows that the plurality of predictions are selected by an 'n' bit index).

36. In regard to claim 8:

37. The combination of Emma et al. in view of McFarling as applied to claim 1 does not explicitly teach that the said selector comprises a saturating up/down counter.

38. However McFarling teaches the use of a saturating up/down counter to select the best predictor to use when using combined predictors of any kind (pg. 12, lines 30-34). Using this scheme improves prediction accuracy significantly as shown in fig. 13.

39. Therefore it would have been obvious to one of ordinary skill in the art to modify the selector by using a saturating up/down counter to perform the selection as McFarling teaches that it improves prediction accuracy.

40. In regard to claim 9:

41. The combination of Emma et al. in view of McFarling as applied to claim 8 teaches the apparatus of claim 8, wherein said saturating up/down counter stores a selection value from among one of: strongly first predictor, weakly first predictor, weakly second predictor, and strongly second predictor (McFarling: pg. 12, lines 33-37 and the table indicate this limitation inherently. This is because a 2-bit saturating up/down counter has 4 states and according to the table, the counter is incremented when the first predictor is correct and the second is wrong and decremented when the first predictor is wrong and the second predictor is correct, the saturating counter would inherently store one of strongly first predictor (11), weakly first predictor (10), weakly second predictor (01), and strongly second predictor (00)).

42. In regard to claim 10:

43. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 1, further comprising: a register, coupled to said second predictor, for storing said global branch history (McFarling: fig. 10 shows a 'GR' register).

44. In regard to claim 11:

45. The combination of Emma et al. in view of McFarling as applied to claim 1 teaches the apparatus of claim 10, wherein said register comprises an N-bit shift register for storing N previous outcomes of whether branch instructions executed by the microprocessor where taken or not taken (McFarling: pg. 6, lines 27-30).

46. Claims **12-21, 23, 24, 30-36, 40-44** are rejected under 35 U.S.C. 103(a) as being unpatentable over Black et al. (US005761723A) in view of McFarling ("WRL Technical Note TN-36, Combining Branch Predictors," Digital Equipment Corp., 1993).

47. In regard to claim 12:

48. Black et al. teach a speculative branch prediction apparatus in a pipelined microprocessor (fig. 3) having an instruction cache (14), the instruction cache receiving a fetch address on an address bus for selecting an instruction in the instruction cache (fetch address stored in IFAR 44 is shown to be sent to the instruction cache 14 over an address bus), a branch instruction presumably present in the cache line, the apparatus comprising:

a speculative branch history table (BHT 50 is speculative because it uses the fetch address to make its prediction before it is known whether a branch exists at the location or not), for providing a first direction prediction of the branch instruction;

a speculative branch target address cache (BTAC 48 is speculative because it uses the fetch address to make its prediction before it is known whether a branch exists at the location or not), coupled to the address bus (shown coupled to address bus connected to IFAR 44), for providing a second direction prediction of the branch instruction, and for providing a selection for selecting between said first and second direction predictions (The BTAC outputs a HIT/MISS signal which is used in the selection of the first and second direction predictions [col. 8, lines 51-54, 58-66]); and a multiplexer (40), coupled to said BHT (via decode buffer 52 and decode prediction 54) and said BTAC, for selecting one of said first and second direction predictions based on said selection (based on the HIT/MISS signal and other signals, the mux 40 is used to select the target address corresponding to the first and second direction predictions [col. 8, lines 51-66]);

wherein said second prediction is provided in response to the fetch address even though the branch instruction may not be present in the instruction cache line (BTAC 48 uses the fetch address stored in IFAR 44 to make its prediction as shown in fig 3. Therefore this prediction is done before it is known whether a branch exists at the cache line or not).

49. Black et al. do not explicitly mention that the fetch address is used for selecting a cache **line** in the instruction cache. Also, Black et al. teach a selection mechanism that is static based on priority (col. 8, lines 65-66) and not a **selection prediction** as in the current invention.

50. “Official Notice” is taken that it is well known and expected in the art to select a cache line from a cache based on a fetch address in order to receive not only the data at the particular address required but the data near to it also to take advantage of the principle of locality while fetching.

51. McFarling teaches to provide a selection prediction for selecting from among a plurality of predictors by the use of a saturating up/down counter to select the best predictor to use when using combined predictors of any kind (pg. 12, lines 30-34). Using this scheme improves prediction accuracy significantly as shown in fig. 13.

52. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor to fetch a cache line in response to a fetch address and modify the BTAC 48 and multiplexor 40 of Black et al. by allowing for the BTAC to provide for a selection prediction to select the better predictor based on past performances indicated by the saturating counters and replacing the static priority-based selection policy with the dynamic prediction-based policy and providing the multiplexor 40 with the selection prediction to make the selection between the predictors.

53. One would have been motivated to have made this modifications because by fetching an entire cache line, one can take advantage of the prefetching instructions into the processor before actually addressing them and further, a dynamic prediction based selection scheme proves to be an efficient selection mechanism as taught by McFarling. By using the a dynamic selection scheme in stead of the static priority-based scheme, one would be able to prevent occurrences when a previous stage actually provides the correct prediction and the succeeding stage provides an incorrect prediction leading to large misprediction penalties.

54. In regard to claim 13:

55. The combination of Black and McFarling as applied to claim 12 does not explicitly teach a global branch history register, coupled to said BHT, for storing a global history of directions of branch instructions previously executed by the microprocessor.

56. However, McFarling teaches that a more efficient prediction might be made using both the branch address and the global history (pg. 9, lines 26-28). The global history is the history of the directions of branch instructions previously executed by the microprocessor (pg. 6, lines 29-30). He introduces the gshare predictor that uses the exclusive ORing of the branch address and global history stored in a global history register ('GR' fig. 10) to index the branch history table (pg. 11, lines 31-32) and shows that it has the better prediction capabilities than other global history schemes in most cases (pg. 11, lines 33-35, fig. 11). Also he suggests combining different prediction schemes in order to attain better prediction accuracy (pg. 16, lines 34-35).

57. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the BHT presented by Black et al. by adding exclusive-OR circuitry and a global history register so as to index the BHT with the exclusive OR of the fetch address and global history instead just the fetch address.

58. One would have been motivated to do so because it would have improved the prediction accuracy of the prediction means and hence improved the performance of the microprocessor by having less number of branch misprediction stalls.

59. In regard to claim 14:

60. The combination of Black and McFarling as applied to claim 13 teaches that said BHT provides said first direction prediction in response to a function (exclusive-OR) of the instruction cache fetch address and said global history stored in said global branch history register (McFarling: pg. 11, lines 31-32).

61. In regard to claim 15:

62. The combination of Black and McFarling as applied to claim 13 teaches the apparatus of claim 14, wherein said function comprises a logical exclusive OR of said global history stored in said global branch history register and a portion of the instruction cache fetch address (McFarling: pg. 11, lines 31-32).

63. In regard to claim 16:

64. The combination of Black and McFarling as applied to claim 13 teaches the apparatus of claim 14, wherein said BHT comprises an array of storage elements for storing a plurality of direction predictions (Black: col. 6, lines 53-56), wherein said array is indexed by said function of the instruction cache fetch address and said global history (McFarling: pg. 11, lines 31-32).

65. Claim **17** is rejected under 35 U.S.C. 103(a) as being unpatentable over Black et al. (US005761723A) in view of McFarling ("WRL Technical Note TN-36, Combining Branch Predictors," Digital Equipment Corp., 1993) as applied to claim 12 in further view of Emma et al. (US005353421A).

66. In regard to claim 17:

67. The combination of Black and McFarling as applied to claim 13 does not explicitly teach the apparatus of claim 16, wherein each of said storage elements is configured to store a plurality of direction predictions for selection as said first direction prediction.

68. Emma et al. teach that multiple branches may exist in an instruction fetch segment (cache line) (col.12, lines 62-65). Hence each entry in the BHT of Emma et al. has a plurality of direction predictions for selection (fig. 9, segment entry information 82 has plurality of direction predictions "T" for each branch).

69. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the BHT of Black et al. by adding a plurality of direction predictions for each branch in the cache line.

70. One would have been motivated to do so because as a cache line is being fetched from the instruction cache, there may be a plurality of branches in it that need to be predicted as taught by Emma et al. and in order to increase that chances of an accurate prediction, a plurality of direction predictions would be required.

71. In regard to claim 18:

72. The combination of Black and McFarling as applied to claim 12 teaches the apparatus of claim 12, wherein each of said first direction prediction, said second direction prediction, and said selection prediction comprises a plurality of predictions (each of the BHT, BTAC, and the selection prediction comprise of a plurality of predictions as shown by the plurality of prediction entries in each (Black: col. 6, lines 53-55, 25-28; McFarling: pg. 12, lines 32-33, fig. 12)).

73. In regard to claim 19:

74. The combination of Black and McFarling as applied to claim 12 teaches the apparatus of claim 18, wherein said multiplexer selects one of said plurality of predictions for each of said first and second direction predictions in response to a corresponding one of said plurality of selection predictions.

75. In regard to claim 20:

76. The combination of Black and McFarling as applied to claim 12 teaches the apparatus of claim 19, further comprising:

control logic (Black: IFAR register 44), coupled to said multiplexer (fig. 3), for receiving said one of said plurality of predictions (predicted target addresses) for each of said first and second direction predictions from said multiplexer, said control logic configured to cause the microprocessor to selectively speculatively branch or not branch based on said one of said plurality of predictions (Black: The IFAR register holds the next fetch address [col. 5, lines 61-62]. From fig. 3 one can see that depending on the predictions the IFAR register 44 may receive the speculative target address of a branch speculatively predicted by the BTAC 48 causing the processor to speculatively branch or the next sequential address from the sequential address calculator 46 causing the processor to not branch).

77. In regard to claim 21:

78. The combination of Black and McFarling as applied to claim 12 teaches the apparatus of claim 20, wherein said control logic (Black: IFAR register 44) is configured to cause the microprocessor to selectively speculatively branch to a speculative branch target address provided by said BTAC in response to the fetch address (Black: fig. 3 shows that the IFAR 44 will receive the speculative target provided by BTAC 48 when the BTAC address is selected to be outputted by the multiplexer 40 causing the processor to speculatively branch).

79. In regard to claim 23:

80. Black et al. teach a microprocessor (fig. 1) for speculatively branching, comprising:

an instruction cache (fig. 1, 14), for providing instruction bytes selected by said fetch address provided on an address bus (fig .3, fetch address stored in IFAR 44 is shown to be sent to the instruction cache 14 over an address bus to select the instruction bytes);

a speculative branch history table (BHT 50 is speculative because it uses the fetch address to make its prediction before it is known whether a branch exists at the location or not), coupled to said address bus (fig. 3), for providing a first prediction of whether a branch instruction that is presumed to be present in said instruction cache line will be taken (BHT provides taken/not taken prediction based on the history bits stored [col. 6, lines 53-55, col. 7, 6-8]);

a speculative branch target address cache (BTAC 48 is speculative because it uses the fetch address to make its prediction before it is known whether a branch exists at the location or not), coupled to said address bus (fig. 3), for providing a second prediction of said presumed branch instruction (BTAC provides a predicted target address [col. 6, lines 4-5]) and for providing a selector (The BTAC provides a HIT/MISS signal to the multiplexer 40 via the address selector 42 which is used in the selection of the first and second direction predictions [col. 8, lines 51-54, 58-66]); and

control logic (fig .3, IFAR register 44), coupled to said BHT (via decode buffer 52, decode prediction 54, address selector 42, and MUX 40) and BTAC (via MUX 40), for causing the microprocessor to speculatively branch if one of said first and second predictions selected by said selector predicts that said presumed branch instruction will be taken (The IFAR register holds the next fetch address [col. 5, lines 61-62]. From fig. 3 one can see that depending on the predictions selected by the MUX 40, the IFAR register 44 may receive the target address of a branch speculatively predicted by the BTAC 48 or BHT 50 causing the processor to speculatively branch).

81. However, Black et al. do not explicitly mention that the fetch address is used for selecting a cache **line** to be provided by the instruction cache. Also, Black et al. do not explicitly teach that first prediction is provided based on a combination of said fetch address and a global branch history.

82. "Official Notice" is taken that it is well known and expected in the art to select a cache line from a cache based on a fetch address in order to receive not only the data at the particular address required but the data near to it also to take advantage of the principle of locality while fetching.

83. McFarling teaches that a more efficient prediction might be made using both the branch address and the global history (pg. 9, lines 26-28). The global history is the history of the directions of branch instructions previously executed by the microprocessor (pg. 6, lines 29-30). He introduces the gshare predictor that uses the exclusive ORing of the branch address and global history stored in a global history register ('GR' fig. 10) to index the branch history table (pg. 11, lines 31-32) and shows that it has better prediction capabilities than other global history schemes in most cases (pg. 11, lines 33-35, fig. 11). Also he suggests combining different prediction schemes in order to attain better prediction accuracy (pg. 16, lines 34-35).

84. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor to fetch a cache line in response to a fetch address and modify the BHT presented by Black et al. by adding exclusive-OR circuitry and a global history register so as to index the BHT with the exclusive OR of the fetch address and global history instead just the fetch address.

85. One would have been motivated to have made this modifications because by fetching an entire cache line, one can take advantage of the prefetching instructions into the processor before actually addressing them and by using the gshare predictor one would have improved the prediction accuracy of the prediction means and hence improved the performance of the microprocessor by having less number of branch misprediction stalls.

86. In regard to claim 24:

87. The combination of Black et al. in view of McFarling as applied to claim 23 teaches the microprocessor of claim 23, wherein said control logic (Black: IFAR register 44) causes the microprocessor to speculatively branch to a speculative branch target address provided by said BTAC based on said fetch address (Black: The IFAR register holds the next fetch address [col. 5, lines 61-62]. From fig. 3 one can see that when the BTAC is selected by the MUX 40, the IFAR register 44 receives the speculative target address of a branch speculatively predicted by the BTAC 48 based on the fetch address from IFAR 44 causing the processor to speculatively branch).

88. Claims **25-29** are rejected under 35 U.S.C. 103(a) as being unpatentable over Black et al. (US005761723A) in view of McFarling ("WRL Technical Note TN-36, Combining Branch Predictors," Digital Equipment Corp., 1993) as applied to claim 23 in further view of Shiell et al. (US005850543A).

89. In regard to claim 25:

90. The combination of Black et al. in view of McFarling as applied to claim 23 does not teach the limitations of a speculative call/return stack, coupled to said BTAC, for storing a plurality of speculative return addresses; wherein said control logic causes the microprocessor to speculatively branch to one of said plurality of speculative return addresses provided by said speculative call/return stack based on said fetch address.

91. However, Shiell et al. teach a call/return stack (fig. 2, 55), coupled to a Branch Target Buffer (fig .2, BTB 56), storing multiple speculative return addresses (col. 9, lines 51-55). When the fetch address for a return from subroutine is fetched, the return address is popped from the stack and speculatively executed by the microprocessor (col. 10, lines 7-14). Also History bits [fig. 3, HIS field] in the BTB provide indication of a call instruction when set to '011' and a return instruction when set to '010' [col. 8, 57-67]).

92. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the BTAC of Black et al. by adding a call/return stack, storing a plurality of speculative return addresses and modifying the control logic so as to speculatively branch to a speculative return address when the address of a return from

subroutine instruction is fetched. Also a field indicating whether the instruction is a branch, call, or return should be added in the BTAC so that it can be known whether to output the target address from the BTAC or the call/return stack.

93. One would have been motivated to make these modifications because by using a call/return stack instructions after a return from subroutine instruction can also be speculatively fetched and executed hence leading to better performance.

94. In regard to claim 26:

95. The combination of Black et al. in view of McFarling further in view of Shiell et al. as applied to claim 25 teaches that the BTAC is configured to provide an indication of whether said presumed branch instruction is a return instruction (HIS bits [fig. 3] in the BTB provide indication of a return instruction when set to '010' [col. 8, 57-67]).

96. In regard to claim 27:

97. The combination of Black et al. in view of McFarling further in view of Shiell et al. as applied to claim 25 teaches the microprocessor of claim 26, wherein said control logic causes the microprocessor to speculatively branch to said one of said plurality of speculative return addresses only if said indication indicates said presumed branch instruction is a return instruction (Shiell: With respect to the 2 sections of instruction code in col. 11, after a CALL is executed, the contents of the BTB and return stack are shown in fig. 4e. The entry with the HIS field set to '010' indicates a return instruction

causing the microprocessor to speculatively fetch the return address by popping it from the stack [col. 13, lines 54-67]).

98. In regard to claim 28:

99. The combination of Black et al. in view of McFarling further in view of Shiell et al. as applied to claim 25 teaches the microprocessor of claim 27, wherein said BTAC is configured to provide an indication of whether said presumed branch instruction is a call instruction (HIS bits [fig. 3] in the BTB provide indication of a call instruction when set to '011' [col. 8, 57-67]).

100. In regard to claim 29:

101. The combination of Black et al. in view of McFarling further in view of Shiell et al. as applied to claim 25 teaches the microprocessor of claim 28, wherein said control logic causes said one of said plurality of speculative return addresses to be pushed onto said speculative call/return stack if said indication indicates said presumed branch instruction is a call instruction (Shiell: col. 13, lines 16-21, 54-55 disclose that on execution of a call instruction that is indicated by HIS field '011', causes the speculative return address to be stored in the stack i.e. pushed).

102. In regard to claim 30:

103. The combination of Black et al. in view of McFarling as applied to claim 23 does not disclose the limitation of said selector being updated in response to a resolved direction of whether said presumed branch instruction is taken.

104. McFarling teaches to provide a selector for selecting from among a plurality of predictors by the use of a saturating up/down counter to select the best predictor to use when using combined predictors of any kind (pg. 12, lines 30-34). On knowing which predictor was correct and which was wrong, the counter is updated according to the table on pg. 12. Using this scheme improves prediction accuracy significantly as shown in fig. 13.

105. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the address selector 42 and multiplexor 40 of Black et al. by allowing for the address selector to select the better predictor based on past performances of the predictor indicated by the saturating counters and replacing the static priority-based selection policy with the dynamic prediction-based policy and providing the multiplexor 40 with the selection prediction to make the selection between the predictors. The saturating counters reflect the past predictions by being updated based on resolved directions.

106. One would have been motivated to have made this modifications because a dynamic prediction based selection scheme proves to be an efficient selection mechanism as taught by McFarling. By using the a dynamic selection scheme instead of the static priority-based scheme, one would be able to prevent occurrences when a

previous stage actually provides the correct prediction and the succeeding stage provides an incorrect prediction leading to large misprediction penalties.

107. In regard to claim 31:

108. The combination of Black et al. in view of McFarling as applied to claim 30 teaches the microprocessor of claim 30, wherein said selector is updated in response to said resolved direction if a selected one of said first and second predictions is incorrect, and if a non-selected one of said first and second predictions is correct (McFarling: table on pg. 12).

109. In regard to claim 32:

110. The combination of Black et al. in view of McFarling as applied to claim 30 teaches the microprocessor of claim 31, wherein said selector is updated by toggling said selector (McFarling: table on pg. 12 indicates that the bits will be toggled when updated).

111. In regard to claim 33:

112. The combination of Black et al. in view of McFarling as applied to claim 30 teaches the microprocessor of claim 31, wherein said selector is updated by counting said selector toward said non-selected prediction (McFarling: table on pg. 12).

113. In regard to claim 34:

114. The combination of Black et al. in view of McFarling as applied to claim 23 teaches the microprocessor of claim 23, wherein said BHT comprises an array of storage elements, for storing a branch history for each of a plurality of branch instructions (col. 6, lines 53-55).

115. In regard to claim 35:

116. The combination of Black et al. in view of McFarling as applied to claim 23 teaches the microprocessor of claim 34, wherein said branch history for each of said plurality of branch instructions comprises a taken/not taken bit (col. 6, lines 53-55 indicate the use of 2-bits to indicate taken/not taken condition. Hence each one is a taken/not taken bit).

117. In regard to claim 36:

118. The combination of Black et al. in view of McFarling as applied to claim 23 teaches the microprocessor of claim 34, wherein said branch history for each of said plurality of branch instructions comprises a saturating up/down counter (col. 6, lines 53-55, col. 7, lines 11-14).

119. In regard to claim 40:

120. Black et al. teach a method for speculatively branching in a microprocessor, the method comprising:

generating first (BHT 50, col. 6, lines 53-56, col. 7, lines 6-8) and second (BTAC 48, HIT/MISS essentially signals taken/not taken, col. 6, lines 30-36) predictions of whether a branch instruction will be taken or not taken, in response to first (col. 6, lines 56-57) and second binary functions (col. 6, lines 28-30) of an instruction cache fetch address;

selecting one of said first and second predictions as a final prediction (Address selector 42 and Multiplexer 40 select among HIT/MISS from BTAC and Decode Correction from BHT as a final prediction by outputting the target address corresponding to that prediction, col. 8, lines 51-58);

and speculatively branching the microprocessor if said final prediction specifies said branch instruction will be taken (if the final prediction of taken is from the BTAC or BHT then the processor will speculatively branch because the BTAC and BHT make speculative predictions based on the fetch address);

wherein said generating, said selecting, and said speculatively branching are performed whether or not said branch instruction is selected by said fetch address (The BHT and BTAC both make their predictions based on the fetch address (col. 6, lines 56-57; col. 6, lines 28-30) before it is known whether the instruction being fetched is a branch or not. Hence, as the selection and the branching are based on these speculative predictions, they are performed whether or not said branch instruction is selected by said fetch address).

121. Black et al. do not explicitly teach that an instruction cache line is selected by the fetch address and that the said selecting is performed in response to a third binary function of said fetch address.

122. "Official Notice" is taken that it is well known and expected in the art to select a cache line from a cache based on a fetch address in order to receive not only the data at the particular address required but the data near to it also to take advantage of the principle of locality while fetching.

123. McFarling teaches to provide a selector for selecting from among a plurality of predictors by the use of a saturating up/down counter to select the best predictor based on the fetch address (fig. 12 shows that the PC is used to index the array of counters) to use when using combined predictors of any kind (pg. 12, lines 30-34). On knowing which predictor was correct and which was wrong, the counter is updated according to the table on pg. 12. Using this scheme improves prediction accuracy significantly as shown in fig. 13.

124. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor to fetch a cache line in response to a fetch address and modify the address selector 42 and multiplexor 40 of Black et al. by allowing for the address selector to select the better predictor based on past performances of the predictor indicated by the saturating counters and replacing the static priority-based selection policy with the dynamic prediction-based policy and

providing the multiplexor 40 with the selection prediction to make the selection between the predictors. The saturating counters reflect the past predictions by being updated based on resolved directions and are indexed by the fetch address.

125. One would have been motivated to make these modifications because by fetching an entire cache line, one can take advantage of the prefetching instructions into the processor before actually addressing them and a dynamic prediction based selection scheme proves to be an efficient selection mechanism as taught by McFarling. By using the a dynamic selection scheme instead of the static priority-based scheme, one would be able to prevent occurrences when a previous stage actually provides the correct prediction and the succeeding stage provides an incorrect prediction leading to large misprediction penalties.

126. In regard to claim 41:

127. The combination of Black et al. and McFarling as applied to claim 40 does not explicitly teach the method of claim 40, wherein said first and second functions are different (both functions are a direct function of the fetch address).

128. McFarling teaches that a more efficient prediction might be made using both the branch address and the global history (pg. 9, lines 26-28). The global history is the history of the directions of branch instructions previously executed by the microprocessor (pg. 6, lines 29-30). He introduces the gshare predictor that uses the exclusive ORing of the branch address and global history stored in a global history register ('GR' fig. 10) to index the branch history table (pg. 11, lines 31-32) and shows

that it has better prediction capabilities than other global history schemes in most cases (pg. 11, lines 33-35, fig. 11). Also he suggests combining different prediction schemes in order to attain better prediction accuracy (pg. 16, lines 34-35).

129. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the BHT presented by Black et al. by adding exclusive-OR circuitry and a global history register so as to index the BHT with the exclusive OR of the fetch address and global history (different function from second prediction function) instead just the fetch address.

130. One would have been motivated to make this modification because by using the gshare predictor one would have improved the prediction accuracy of the prediction means and hence improved the performance of the microprocessor by having less number of branch misprediction stalls.

131. In regard to claim 42:

132. The combination of Black et al. and McFarling as applied to claim 40 does not explicitly teach the method of claim 40, wherein said second binary function comprises a binary function of said fetch address and a global branch history.

133. McFarling teaches that a more efficient prediction might be made using both the branch address and the global history (pg. 9, lines 26-28). The global history is the history of the directions of branch instructions previously executed by the microprocessor (pg. 6, lines 29-30). He introduces the gshare predictor that uses the exclusive ORing of the branch address and global history stored in a global history

register ('GR' fig. 10) to index the branch history table (pg. 11, lines 31-32) and shows that it has better prediction capabilities than other global history schemes in most cases (pg. 11, lines 33-35, fig. 11). Also he suggests combining different prediction schemes in order to attain better prediction accuracy (pg. 16, lines 34-35).

134. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the BHT presented by Black et al. by adding exclusive-OR circuitry and a global history register so as to index the BHT with the exclusive OR of the fetch address and global history instead just the fetch address.

135. One would have been motivated to make this modification because by using the gshare predictor one would have improved the prediction accuracy of the prediction means and hence improved the performance of the microprocessor by having less number of branch misprediction stalls.

136. In regard to claim 43:

137. The combination of Black et al. and McFarling as applied to claim 42 teaches the method of claim 42, wherein said second binary function comprises an exclusive OR of at least a portion of said fetch address and said global branch history.

138. In regard to claim 44:

139. The combination of Black et al. and McFarling as applied to claim 40 teaches the method of claim 40, wherein said first and third binary functions are the same (both are direct functions of the fetch address).

140. Claim **45** is rejected under 35 U.S.C. 103(a) as being unpatentable over Black et al. (US005761723A) in view of McFarling ("WRL Technical Note TN-36, Combining Branch Predictors," Digital Equipment Corp., 1993) as applied to claim 40 in further view of Keller et al. (US006502185B1).

141. In regard to claim 45:

142. The combination of Black et al. and McFarling as applied to claim 44 teaches the method of claim 44, wherein said third binary function comprises a predetermined number of least significant bits of said fetch address (McFarling: shows in fig. 12 the use of the entire PC which comprises of all (a predetermined number) of the least significant bits).

143. However, Black et al. does not teach that the second binary function comprises a predetermined number of the least significant bits of the fetch address.=

144. Keller et al. teaches using the least significant bits of the fetch address to index the Branch Target Cache 18B (col. 15, lines 2-6).

145. "Official Notice" is taken that it is well known and expected in the art that the least significant bits of the fetch address are more random and varying than the most significant bits.

146. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify to the second binary function to comprise of the least significant bits of the fetch address.

147. One would have been motivated to do so because by using the more varying least significant bits of the fetch address there would be less aliasing. Aliasing occurs when 2 or more branches map to the same entry in the BTAC because only a portion of the fetch address is used to index into the BTAC. As branches that are close to each other have similar most significant bits, they would map to the same prediction entry in the BTAC resulting in bad prediction. If the least significant bits are used then the branches would probably map to different entries translating in better prediction.

Conclusion

148. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty, which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections. See 37 CFR § 1.111.

a. Po-Yung Chang et al. ("Alternative Implementation of Hybrid Branch Predictors", IEEE, Proceedings of Microarchitecture-28, 1995, pp. 252-257) teaches various implementations of hybrid branch predictors.

b. Yeh and Patt, ("Alternative Implementations of Two-Level Adaptive Branch Prediction", ISCA-19, pp. 124-134, 1992), teach in section 2.2, different implementations of two-level adaptive branch prediction including GAg, PAg, and PAp (in which there are multiple predictions per address which are selected based on the history).

c. Chang (US005687360) teaches a hybrid prediction scheme.

d. Liu et al. (US006088793A) teaches multiple speculative prediction schemes in a single processor.

e. Hoyt et al. (US005812839A) teaches a hybrid prediction scheme where the BTB receives speculative branch prediction information from a second prediction mechanism using the fetch address. It also shows a return stack buffer.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amol V. Gole whose telephone number is 703-305-8888. The examiner can normally be reached on 9:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 703-305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AVG
amol.gole@uspto.gov

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100